

目視での評価に特化した学生のプログラムの品質評価システムの提案と試作 A Proposal of System for Quality Evaluation of Students' Source Code Focused on Visual Assessment

岡本 八仁[†]
Hachijin Okamoto

杉田 基樹[‡]
Motoki Sugita

早川 智一[†]
Tomokazu Hayakawa

1. はじめに

近年、大学などの教育機関ではプログラミングの実習が行われているが、学生のプログラムの評価では、実行結果だけでなく実装方法の正しさの評価が求められる。なぜならば、学生のプログラムは、実行結果が正しくても実装が誤っている場合がある [1] ためである。

学生の実装の正確な評価を自動で行う手法は提案されている [2] [3] が、時には、教員の目視による評価が必要になる場合がある。これは、課題によって実装の評価基準が異なり、それに対応する自動評価ツールが存在するとは限らないためである。

しかし、実装の目視評価は、教員に大きな負担を強いするという課題がある。これは、実習課題の数や履修者数に応じて必要な工数が増大するためである。

プログラムの評価における教員の負担はできる限り小さい方が望ましい。これは、教員の余剰工数を学生とのコミュニケーションや教材の高品質化に充てることで、学生の学びが増え、実習の質が向上するためである。

しかし、実装の目視評価が教員に負担を強いる具体的な要因を明らかにし、同評価の効率化を試みた研究は我々の調査の限り存在しない。

本論文の目的は、既存の手法における実装の目視評価が教員にとって大きな負担となる要因を明らかにし、より少ない負担での目視評価を可能にすることにある。我々は、既存の手法における目視評価の課題を明らかにするため、事前調査 (3 章) を行った。その結果、ファイルの開閉やタブ・ウィンドウの切り替えが、目視評価の効率を低下させ、負担を大きくすることがわかった。そこで我々は、単一画面でファイルをシームレスに切り替えながらソースコードの目視と評価結果の記録が可能なシステムを提案し、タブ・ウィンドウの切り替えやファイルの開閉なしでの目視評価の実現を試みる。

本論文の構成は次のとおりである。2 章では関連技術・研究を紹介する。3 章では事前調査の結果を報告する。4 章では提案手法を概説する。5 章では設計と実装を概説する。6 章では評価結果を報告する。7 章では今後の展望を述べる。

2. 関連技術・関連研究

テキストエディタ (以下、エディタ) は、テキストファイルの閲覧・編集を行うツールである。シンタックスハイライト機能やファイルの一覧表示機能をもつ場合があり、ソースコードの目視の用途に適している。

コードレビューツールは、ソフトウェア開発工程におけるコードレビューを支援するツールである。GitHub [4] などのソースコード管理ツールにも内蔵されており、コードの特定の部分にコメントする機能を持つため、評価結果の記録の用途に適している。

堀越ら [5] は、プログラミング学習環境 Bit Arrow の採点支援として、学習者のソースコードと入出力テストの結果を一覧表示可能な採点フォームを生成する手法を提案している。この研究は、実装の評価にかかる教員の工数を削減するという目的が本研究と共通しているが、目視での評価を対象としていない点で異なる。

3. 事前調査

我々は、既存の手法での目視評価の課題を明らかにするため、本研究室の学生 11 名にアンケートを行った。

まず、実装の目視評価を行う場合にどのツールを用いるかを質問した結果、6 名がソースコードの目視にエディタ、評価結果の記録に表計算ソフトを使用すると回答した。他の回答は、エディタとコードレビューツールが 1 名、エディタのみが 3 名、コードレビューツールのみが 1 名であった。各ツールの使用理由として、エディタはシンタックスハイライトがあること、表計算ソフトは記録が容易であること、コードレビューツールは行ごとにコメントが可能であることが挙げられた。

次に、前問の回答にかかわらず、既存のツールを用いる際にどのような点が不便に感じるかを質問した。その結果、エディタに対して 9 名がファイル開閉、8 名がタブ・ウィンドウの切り替えの煩わしさを指摘した。また、コードレビューツールに対して 8 名がページの切り替え、プログラムの実行結果の確認の煩わしさを指摘した。

我々は、以上の調査から、ファイルの開閉やタブ・ウィンドウの切り替えの煩わしさを解消することで、実装の目視評価が効率化し、教員の負担を減らせると考えた。

4. 提案手法

提案システムの画面を図 1 に示す。画面は次の要素で構成されている：(1) ファイル検索部分；(2) ファイル

[†]明治大学理工学部 School of Science and Technology, Meiji University

[‡]明治大学大学院理工学研究科 Graduate School of Science and Technology, Meiji University

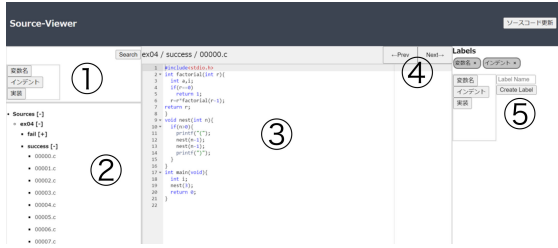


図 1 提案システムの画面

覧；(3) ソースコード閲覧部分；(4) ファイル切り替えボタン；(5) ラベル管理部分。単一画面で作業が完結するため、タブ・ウィンドウの切り替えが不要な特徴をもつ。

4.1 ソースコードの閲覧・切り替え

提案システムは、既存のエディタと同様にシンタックスハイライト機能をもつ。また、閲覧ファイルを切り替える方法として、一覧から選択する他に、ソースファイル切り替えボタンで前後のファイルへ移動できる。これにより、何件ものプログラムを次々に閲覧する場合のファイル開閉の煩わしさを軽減している。

4.2 ソースファイルのラベリング

提案システムはラベリングによる評価結果の記録が可能である。例えば、「変数名が不適切」・「非効率な実装方法」などのラベルを作成し、ソースファイルに付与できる。ラベリングを用いた理由としては、(1) 記録用のファイルを用意する必要が無い点、(2) 記録後の分析（絞り込みなど）がしやすい点——が挙げられる。

5. 設計・実装

我々は、提案システムの構成として、SPA (Single Page Application) 形式の Web アプリケーションを採用した。これは、SPA が、単一のページのみで構成され、ページ遷移を伴わずに高速で画面を更新できるため、ソースファイルをよりシームレスに切り替えられるためである。本論文では、Web サーバの実装に Node.js (Ver. 18.17.1)、確認画面の実装に Vue.js (Ver. 3.4.21)、ラベル情報を保存するデータベースに SQLite (Ver. 5.1.7) を用いた。

6. 評価

本論文では、提案システムが教員の負担を軽減するかを評価するため、既存のツールと提案システムとでの実装の目視評価の作業時間を比較した。

6.1 作業対象のソースコード

提案システムの評価には、本学科の C 言語の実習「プログラム実習」のソースコードを用いた。これは、プログラム実習が本学科の 1 年次を対象としており、誤りを含む実装が多く得られると考えたためである。実習課題の難易度は 5 レベルに分かれており、最も難易度の高いレベル 5 の課題に提出されたソースコード 450 件を半分ずつデータ 1 とデータ 2 に分け、目視評価作業に用いた。

表 1 目視評価作業に要した時間の比較

作業者	提案ツールなし	提案ツールあり	短縮率
A-1	123 分 14 秒	81 分 53 秒	33.6%
A-2	63 分 08 秒	44 分 30 秒	29.5%
A-3	153 分 11 秒	85 分 42 秒	44.1%
B-1	69 分 17 秒	59 分 40 秒	13.9%
B-2	78 分 08 秒	75 分 58 秒	2.8%
B-3	69 分 21 秒	57 分 59 秒	16.4%

6.2 評価の流れ

本研究室の学生 6 名を作業者とし、提案システムの評価を行った。作業順序による影響を低減するため、作業者をグループ A・B に分け、既存のツールと提案システムを用いる順序を入れ替えて評価を行った。グループ A は、既存のツールでの作業後に提案システムで作業し、グループ B は逆の順番で作業した。

6.3 評価結果

作業者ごとの作業時間を表 1 に示す。提案システムは、既存のツールでの作業時間を平均約 23.4%短縮した。グループ B は、既存のツールでの作業時間がグループ A に比べ特に短いことから、作業自体への慣れの影響が大きいが推測できる。しかし、作業順序に関わらず全ての作業者の作業を短縮したため、提案システムは教員の負担の軽減に有用であると考えられる。

7. おわりに

本論文では、教員の負担を軽減するため、実装の目視評価を効率化するシステムを提案した。評価の結果、提案システムは教員の負担の軽減に有用であるという結論を得た。提案システムの改善点について作業者に自由回答を募った結果、作業進捗の表示・保存機能や、コードレビューツールのような行ごとのコメント機能を求める意見が挙がった。今後の展望としては、これらの機能などを実現して提案システムを拡充し、教員の負担をより軽減可能なシステムの提案を行う予定である。

参考文献

- [1] 横井翔太, 早川智一: C 言語のプログラムを関数ごとにテスト可能なウェブブラウザ上で動作するインタプリタの提案と試作, 第 84 回全国大会講演論文集, Vol. 2022, No. 1, pp. 323-324 (2022).
- [2] 高野辰之, 宮川治, 小濱隆司: オブジェクト指向プログラミング教育における採点支援システムの開発とその評価, 技術報告 103(2008-CE-096), 東京電機大学大学院情報環境学研究所, 東京電機大学情報環境学部 (2008).
- [3] 漆原宏丞, 本多佑希, 岸本有生, 兼宗進: 抽象構文木を利用したプログラミング理解度採点の試み, 技術報告 16, 大阪電気通信大学, 四天王寺大学, 大阪電気通信大学 (2021).
- [4] GitHub: GitHub, <https://github.com/>.
- [5] 堀越将之, 長島和平, 長慎也, 兼宗進, 並木美太郎: プログラミング学習環境「Bit Arrow」における採点支援機能, 技術報告 9, 明星大学, 東京農工大学, 明星大学, 東京農工大学 (2018).